Managing an Oracle Unified Method (OUM) Project Using Scrum

An Oracle White Paper
February 2011

ORACLE

Managing an Oracle Unified Method (OUM) Project Using Scrum

Managing an Oracle Unified Method (OUM) Project Using Scrum

EXECUTIVE OVERVIEW

The intent of this white paper is to provide you with a high-level guide for managing a Scrum based project using the Oracle Unified Method (OUM) as the method and source of work products. It is not intended to be a comprehensive guide for Scrum nor OUM; rather it introduces some important characteristics of Scrum and presents techniques on how you can add Scrum principles to an OUM project. While focused on applying Scrum on an OUM project, the concepts are relevant to the various other "flavors" of agile methods such as XP and Crystal.

INTRODUCTION

"Scrum hangs all of its practices on an iterative, incremental process skeleton."

—Ken Schwaber
Agile Project Management With Scrum

If you have never managed a project with a Unified Process approach, or are not familiar with the terms "iterative" or "incremental" you should read the "Read Me First" which can be found on the homepage of OUM. In addition, a list of Scrum books and websites is available for you to use in the References and Works Cited section at the end of this white paper.

WHAT IS SCRUM?

Scrum is one of the agile processes and is an adaptive project management approach based on the concept that the individuals developing the software are in a position to most positively improve the development of that software.

Scrum was created in 1993 by Jeff Sutherland. The term "scrum" is borrowed from an analogy used in the 1986 study by Takeuchi and Nonaka (Takeuchi), published in the Harvard Business Review. In the study, the authors equated high-performance, cross-functional teams to the packs formed by Rugby teams.

Scrum is a framework and is independent of any specific engineering practices. Therefore, the Scrum framework can be used in conjunction with other agile approaches, like OUM.

SHORT PRIMER ON SCRUM TERMINOLOGY

Scrum is comprised of three roles, four ceremonies and three artifacts.

Scrum Roles

- Product owner – responsible representing the use population and for the business return on the project.

- ScrumMaster – serves in the role of a facilitator/coach, ensures that the team is functional and productive and removes barriers.
- Team – cross-functional and self-organizing; accountable to get the work done.

Scrum Ceremonies

- Sprint planning – the team meets with the product owner to choose a set of goals to deliver during a sprint.
- Daily scrum – the team meets each day to share progress and barriers. During this meeting, which is timeboxed to 15 minutes each team member answers the following:
    - What did I do yesterday (last working day)?
    - What am I going to do today?
    - What barriers are preventing me from moving forward?
- Sprint reviews – the team demonstrates to the product owner and/or stakeholders the results of what it has completed during the sprint.
- Sprint retrospectives – the team meets to determine for ways to improve the product and the process.

Scrum Artifacts

- Product backlog – the prioritized list of desired project outcomes/features remaining to be implemented in the product.
- Sprint backlog – a subset of work from the product backlog that the team agrees to complete in a sprint, which is divided into tasks, as shown in Table 1 below.

				Work (hours)		
Requirement	Task	Who	Status	work left (ETC)	AC	BAC
Global Search	Database	MM	finished	0	3	4
	Unit testing scripts	EW	working	2	1	4
	Codification	AS	working	4	2	4
	Unit testing	MM	pending	4	0	4
Client subscription	Database	MM	finished	0	3	4
	Unit testing scripts	EW	working	2	1	4
	Codification	AS	working	4	2	4
	Unit testing	MM	pending	4	0	4

Table 1: Sprint Backlog

- Burndown chart – a snapshot of the work remaining (a team can have two charts: one for the sprint and one for the overall project). Burndown charts represent remaining work on a vertical access with deadlines or time frames on the horizontal axis. Figure 1 below shows an example burndown chart.

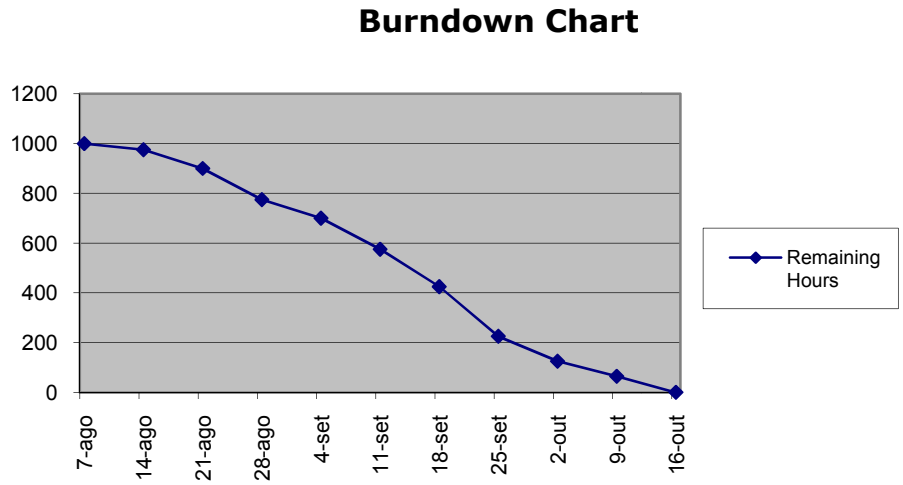

Figure 1: Burndown Chart

SCRUM FRAMEWORK IN A NUTSHELL

An overview of the Scrum framework is shown in Figure 2 below:

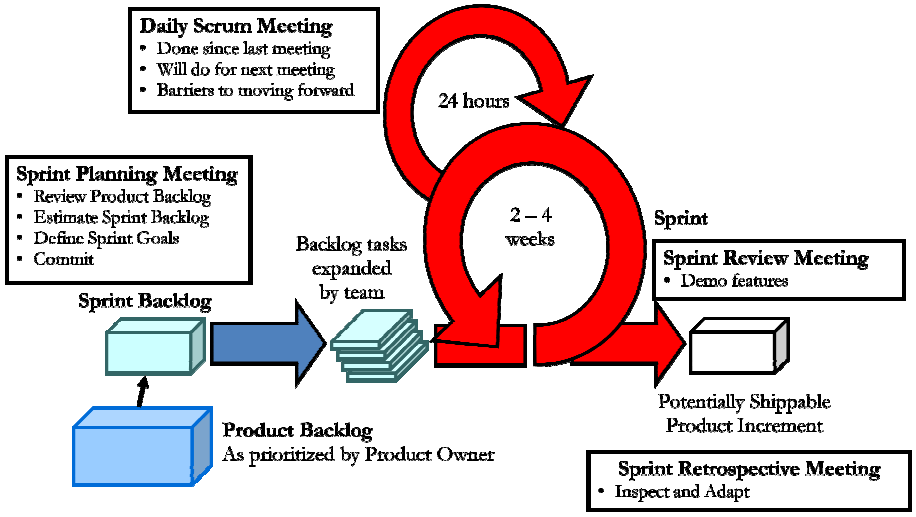

Figure 2: Scrum Framework Skeleton

A product backlog of desired product features is maintained over the course of the project. Items may be deleted or added at any time during the project. It is prioritized by the product owner and items with the highest priority are completed first. It is progressively refined during the course of a project, meaning lower priority items are intentionally course-grained.

Scrum projects progress in a series of sprints, which are kicked off by the sprint planning meeting. During each sprint, typically a two to four week period, the team creates an increment of the software product.

During the sprint planning meeting, it is determined which items from the product backlog will be included in the upcoming sprint. During this meeting, the product owner informs the team of the items in the product backlog that should be

completed. The team then develops estimates and commits to how much of the items they can complete during the next sprint. The selected items, along with the more detailed tasks for completing the requirements, compose the sprint backlog.

During a sprint, the sprint backlog is frozen, which means that the requirements are set for that sprint and cannot be changed. A sprint is usually two to four weeks in length and there can be no changes to the sprint backlog during the sprint. The team meets each day for the daily scrum meeting to assess its progress.

As the sprint progresses, the ScrumMaster coaches the team, removes barriers and keeps the team focused on the goal(s) of the particular sprint. At the end of the sprint, the increment should be potentially shippable, which means it is ready to deliver to a customer, put into production, or demonstrate to the product owner and/or stakeholders.

"This requires that the increment consist of thoroughly tested, well-structured, and well-written code that has been built into an executable and that the user operation of the functionality is documented, either in Help files or in user documentation"

—Ken Schwaber,
Agile Project Management with Scrum

The sprint ends with a sprint review and retrospective. Items from the sprint backlog which are not complete are rolled to a future sprint. The cycle continues until the project objectives are met, the budget is depleted or a deadline is reached. The scrum framework ensures the items with the highest business value are addressed during the course of the project.

OUM DRIVEN BY SCRUM

The scrum framework can be applied during projects which include complex custom software development. OUM uses the Manage Focus Area - Project Management Method (OUM Manage) to provide a framework in which projects can be planned, estimated, controlled, and tracked in a consistent manner.

The following sections discuss how to apply the Scrum framework in the context of the OUM Manage phases.

APPROACH to PROJECT START UP

The Project Start Up phase of OUM Manage is designed to establish the successful start of any OUM project. Therefore on an agile project, the project manager will still be responsible for conducting the necessary start up, establishing the project infrastructure and securing project resources.

Looking at the Project Start Up phase of OUM Manage, you see that the key output of this phase is the Project Management Framework, which is the start of the Project Management Plan (PMP). You may think that in a Scrum project the level of documentation implied in the PMP is not required or is incompatible with an agile approach. However, this is not the case since agile is focused on just enough documentation created just in time to create working software.

On an agile project, as well as any OUM project, documents are created and maintained only to the extent necessary to support the project and the required development. On agile projects, the elements of the PMP are developed in a collaborative environment with the entire team and many PMP elements can be

documented on flip-charts, white boards or in the case of a dispersed team – electronic collaboration tools.

Also, keep in mind that in the Bid Transition process it will be vital to ensure the customer is aware that an agile approach is going to be followed for the project. The contract, whether implicit or explicit, must reflect the fact that in an agile approach, requirements can and will change as more information is uncovered during the course of the engagement.

The customer must understand that scope can be variable in a Scrum project since the time and cost sides of the triangle are often fixed. Figure 3 below represents the "Project Management Triangle," where each side represents a constraint. One side of the triangle cannot be changed without affecting the others.

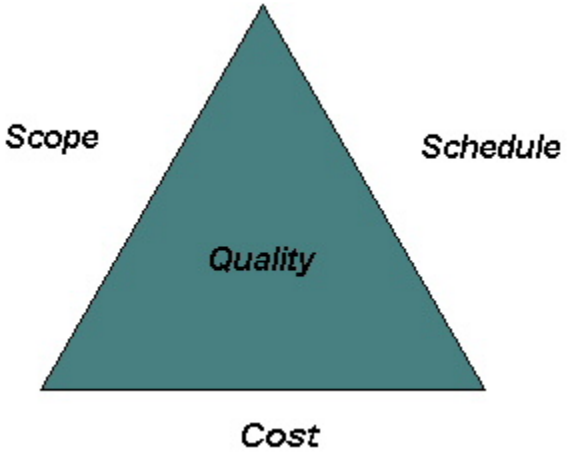

Figure 3: The Project Management Triangle (Triple Constraint)

In the situation where a project runs into many issues, scope can be impacted to the extent that the full functionality expected may not be delivered. Scrum, through its nature, trades off between scope and schedule by freezing schedule and adjusting scope as necessary.

To have a successful Scrum project, at a minimum the customer must recognize the business and cultural benefits of Scrum and agility and be willing to establish the necessary project organizational structure that includes self-managing, dedicated teams of cross-functional resources.

An agile project manager's primary goal is to enable the team to solve its own problems and develop solutions.

Role of Project Manager on a Scrum Project

Agile projects do not require project managers to directly manage the detailed activities of the team and the Scrum process does not list project manager as an official role. Instead, the project manager's main responsibility is to set the team up for success and then provide support to the team from the outside. In some cases, a Scrum Team may include a project administrator to handle communications, project reporting, contract administration, etc.

An agile project manager's primary goal is to enable the team to solve its own problems and develop solutions. Activities such as conducting team health assessments, removing organizational barriers and allowing for failure become central to the agile project manager's role. In the case of an agile project, the project manager essentially becomes a coach to the team. The key skills necessary for any agile project manager are facilitation and collaboration.

APPROACH to PROJECT EXECUTION AND CONTROL

Similar to OUM, Scrum is iterative and incremental. There are components of Scrum and OUM which are comparable and complimentary. The table below discusses how to incorporate Scrum principles and tools in an OUM project.

Comparable Scrum and OUM Components

Scrum	OUM
Sprint	***Iteration***
A sprint consists of a two to four timebox with precise entry and closure points: the sprint planning meeting and the sprint review. Probably the most polemic comparison with OUM would be comparing the sprint with the iteration. However, this is a common misunderstanding in OUM as well. The sprint is somewhat equivalent to a two to four week iteration. Both a sprint/Scrum and an iteration/OUM cover activities of Analysis/Design/Construction/Test/Documentation. However, in a sprint you are supposed to focus on finishing the whole sprint backlog. In the case of the iteration the team may be working on analysis of a subset of use cases while they may also be working on testing another subset of use cases that were developed in the previous iteration. Therefore in OUM, an iteration may be covered by several sprints.	
Product backlog	***MoSCoW***
In Scrum there is a product backlog which is a prioritized list of requirements for the given project. This is equivalent to the MoSCoW list in OUM. Simply stated, the MoSCoW List registers the known business requirements. Each requirement is classified by the first letter of: Must, Should, Could or Won't. For more information on the MoSCoW list, please see OUM task RD.045 – Prioritize Requirements (MoSCoW).	
Sprint backlog	***M's and S's from MoSCoW***
The sprint backlog consists of the items selected by the product owner to be addressed during the current sprint. The sprint backlog is created during the sprint planning meeting. This is equivalent to the top priority items in the MoSCoW list (the Musts and Shoulds) for a given iteration, along with the associated tasks to accomplish during the sprint.	
Sprint Reviews/Retrospectives	***Iteration Reviews***

Sprint and iteration reviews both involve reviewing the product produced during the sprint or iteration and ensuring it meets the quality standards and identifying items to be included in future sprints/iterations. Sprint retrospectives and iteration reviews also include an examination of the project processes and making corrections to improve team efficiency and effectiveness.

Table 2: Scrum and OUM Components

A Note Regarding Uses Cases and Agile Projects

As you know, OUM is business process and use case-driven and therefore makes extensive use of uses cases in capturing, analyzing and tracking requirements. Many experts in the agile community, including Alistair Cockburn, are proponents of use cases over other agile techniques such as user stories since these other techniques tend to be too unstructured, particularly for larger projects. The decision whether to write use cases or utilize another agile requirements management technique should be driven by your customer's preference as well as your own experience and comfort level with use cases versus other techniques.

Reference: Alistair Cockburn, "Why I still use use cases", 19-June-2008 <http://alistair.cockburn.us/Why+I+still+use+use+cases>

On an agile project, as with any other OUM project, it is advisable that use cases be written to the detail necessary to drive out the significant risks and understand the requirements of the system. In general OUM prescribes writing use cases at the user goal level, also known as "sea level".

Scrum Framework Applied to the OUM Implement Focus Area

Scrum and the OUM Implement Focus Area employed together provides the structure needed for robust requirements analysis and definition provided by OUM, while being able to take advantage of Scrum's highly collaborative and agile development characteristics. OUM also provides the guidance for the tasks needed to transition and release software into product, often a complex and challenging endeavor.

Figure 4 below depicts how the Scrum framework is effectively applied on an OUM project.

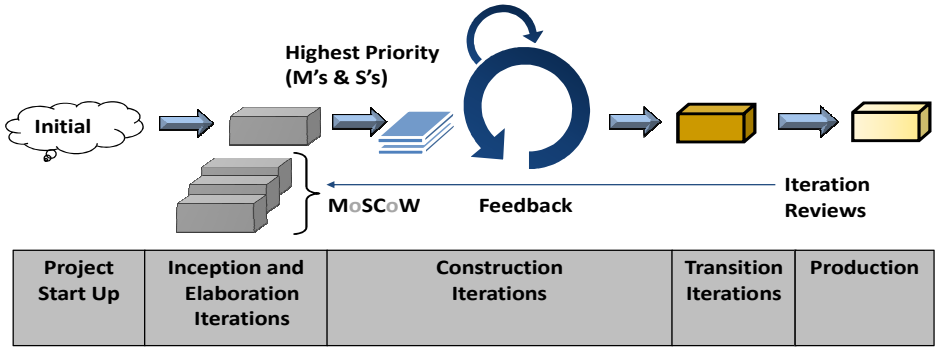

Figure 4: OUM driven by Scrum

Inception and Elaboration

The overriding goal of the Inception phase is to achieve concurrence among all stakeholders on the lifecycle objectives for the project.

At the end of the Elaboration phase, most of the requirements should be analyzed and designed. At this time, the architecture should be stabilized.

Inception and Elaboration are shown on the diagram above in a condensed format. One of the key success factors of any project is the ability of the project team to identify and validate customer requirements. In OUM, requirements are understood at a high-level in the OUM Manage Project Start Up phase. At this early stage, the project direction and goals are understood and requirements are often defined at a coarse-grained level.

We begin the first iteration of the Inception phase with the goal of refining the requirements and nailing down the project scope. Requirements are further refined in the subsequent iterations of the Inception and Elaboration phases and are documented in the MoSCoW list, which as we saw above represents the product backlog in the Scrum framework.

Construction

The goal of the Construction phase is to take the solution from detailed requirements models, through development and testing of components, and integration to a system that is ready for a first release that goes into production, often a limited release and often called a beta release.

The OUM Construction phase is where the heart of the Scrum framework is leveraged. Iterations take the form of two to four week sprints. Each sprint starts with the sprint planning meeting where the highest priority items from the MoSCoW list are selected to be completed by the product owner. Keep in mind that in OUM, the priority of requirements is also determined by complexity, risk and dependencies.

At the end of each sprint, a tested, potentially shippable increment of the product will be produced. Further Construction iterations (sprints) may be required to continue to build additional requirements into the product until the point where the project is ready to move into the Transition phase.

The goal of the Transition phase is to take the completed solution from installation onto the production system through the Acceptance Test to launch of the live application, open and ready for business.

Transition

The OUM Transition phase is comparable to the concept of the Scrum "release sprint" where final testing is performed. A release sprint does not add functionality to the product, rather it is used to apply all of the tasks required to put the product into production. The software product will be updated and refined during the Transition phase as necessary based on the testing results.

The goal of the Production phase is to operate the newly developed system, assess the success of the system and support the users.

Production

Additional requirements may be uncovered while the new system is in Production. These may take the form of change-requests or emergency fixes. The new requirements are added to the MoSCoW list (product backlog) and will be prioritized by the product owner.

OUM Manage Processes Impacted by Scrum

The approach taken to the following OUM Manage processes needs to be altered from the traditional plan-based approach during the course of an agile project:

[SM] Scope Management

The agile strategy is to fix cost and schedule, and then apply effort to implement the highest value/priority features as defined by the customer (the Product Owner in the case of a Scrum project). Therefore in an agile project, scope planning is done as part of release planning. Scope verification and project acceptance is accomplished within the iteration as product features are tested and accepted by owners/customers.

A project manager should give heavy consideration to the dynamic nature of the requirements when developing the Change Management Process and executing scope control. The Change Management Process should be flexible regarding scope but should not necessarily provide automatic approval without regard to risk and the project's budget and schedule constraints.

Scope change control is handled by the management of the product backlog. Assessing and challenging the product backlog on a regular basis is paramount to successful scope change control. The product backlog should be monitored on a regular basis to track how much work is being added to the backlog versus how much is actually getting accomplished during each sprint.

[WM] Work Management

Project planning is a collaborative effort involving the whole team. During the planning meetings, project plans are developed indicating the expected release date and the features to be included. Also during the planning meetings, iteration plans indicating the level of effort in implementing a set of features within the defined timebox are evolved. Managing the work plan involves facilitating the team in their ongoing iterative planning efforts. The plans are then shared in the most visible way possible.

The product burndown chart will provide a view into the project's progress and the team's internal productivity. The chart should show a steady decline in the remaining estimated project effort. Flat lines will indicate a lack of progress possibly due to technical issues, organizational barriers or team productivity deficiencies. Upward moving lines are the first indications of scope creep or may be caused by unforeseen complexities or a lack of team experience. Both a flat and upward trending line should also trigger a re-examination of the plans and estimates created to that point in the project.

"A useful best practice is to *continuously verify quality*, and this not only implies quality of the product, but of the process."

—Craig Larman,
Agile & Iterative Development: A Manager's Guide

[QM] Quality Management

Quality Management is executed throughout an agile project since working code is developed on an incremental basis at a rapid pace. In agile practices, the entire team contributes to defining, maintaining, and enhancing the quality of the evolving product and the iterative process.

Through the daily Scrum Meetings, the team has a means of measuring the results of the iterative project. The team can see how things are working/not working on a day-to-day basis and make the appropriate adjustments.

Quality Management is divided into two processes within OUM – Quality Assurance and Quality Control:

Quality Assurance (QA) is defined in OUM as applying the planned, systematic quality activities and any ongoing processes to ensure that the project employs all delivery processes needed to meet requirements. On an agile project, it is recommended that the team contain dedicated QA resources who participate in decision-making on a daily basis, throughout the entire project. Their input during Inception and Elaboration helps developers write better code and consider more scenarios

Quality Control (QC) is defined in OUM as the process of monitoring specific project results to determine whether they comply with relevant quality standards and identifying ways to eliminate causes of unsatisfactory results. On an agile project, the monitoring is done within the iteration. The entire team is involved in making sure that the features and functionality coded meets the customer's expectations.

The Quality Management Plan will be developed, to the extent necessary, by the entire project team as they determine what tools and technology they will use in writing, running, and reporting tests and results. Product owners must also be involved, since they will be defining and executing the acceptance tests. It is important in Scrum, as in any other agile development process, that the definition of "done" be clearly defined, with inputs from the project team and the product owner, before each sprint.

[STM] Staff Management

As mentioned above, the agile team is a cross-functional self-managing team. Therefore, on an agile project there are neither formal organization charts nor

extensive status reports, rather the team self-manages through regular check-ins on the product and process. At the daily meetings, the team collaboratively examines what they have accomplished so far, examines and analyzes priorities and makes decisions about how to continue. This requires the agile project manager to exercise collaboration and facilitation skills while removing roadblocks to allow the team to function at the highest level possible.

Agile development, due to its iterative nature, implicitly makes risk management a part of the project life cycle. Agile project teams will address risk throughout the project as a part of daily meetings, iteration planning meetings, release planning meetings, and review meetings.

[RKM] Risk Management

The topic of Risk Management is highly debated within the agile community. Members of the agile community disagree on whether explicit risk management is required, Scrum's to manage all types of risk and who should do risk management. The recommendation of OUM (an underlying tenet of which is risk-focused) is that product related risks can be managed effectively within the Scrum. However, risks which are not requirements related and/or are external to the product (issues with vendors or other third-parties, external factors which could cause project delays, etc.) should be escalated to the project manager for resolution.

Agile development, due to its iterative nature, implicitly makes risk management a part of the project life cycle. Agile project teams will address risk throughout the project as a part of daily meetings, iteration planning meetings, release planning meetings, and review meetings.

As part of the product backlog process, the Scrum Team should sort requirements by value and risk and work on the risky requirements long enough to correctly identify and mitigate the risk. The project manager should work with the team to understand which risks cannot be handled through Scrum and facilitate the appropriate action for resolution.

[IPM] Issue and Problem Management

Issues and problems should be handled in a similar manner to risks on an agile project. Issues and problems should be addressed by the entire team. Those issues and problems that are determined to be external to the project team should be escalated to the project manager.

APPROACH TO PROJECT CLOSURE

As with the previous phases, many of the tasks within the Project Closure phase will be a collaborative event for the entire team. As a result, the staff should not be released until the very end of Project Closure. Project Acceptance, since it occurs throughout the life of an agile project, should be a low-ceremony event. The future system enhancements will be stored on the project backlog (MoSCoW list) and can be used to jump-start a future project.

CONCLUSION

OUM and Scrum are different yet similar approaches to solving the same problems with the same objectives.

Scrum is sold as "agile method" in which the team is allowed to work with fewer interruptions, avoiding creating useless documentation (of course keeping the value added documentation) and streamlining the development process avoiding too much management overhead. Because of these characteristics, it is said that Scrum is better for small teams and small systems. Scrum may be used in larger projects with the addition of some more tools, tasks and control.

OUM has characteristics of the Unified Process and DSDM with the addition of leading-practices developed during the course of Oracle Consulting projects. It also offers material for many types and sizes of software development and delivery projects.

Because of this, OUM has a large number of tasks, deliverables and recommendations. It is comparable to a "rodízio" (a typical Brazilian all-you-can-eat restaurant known for the huge variety of options). Like a "rodízio" client, an OUM project manager must choose the tasks that make sense in that specific situation – an intention of eating everything won't end well.

Therefore, OUM project managers should keep project priorities aligned with the client/sponsor and with the team, avoid unnecessary documentation, avoid non-value added tasks and most of all, and facilitate team work.

As with any OUM project, it is recommended that the project manager build the project workplan by starting with the appropriate pre-tailored view for the type of project under consideration (Solution-Driven Application Implementation, Requirements-Driven Application Implementation, etc.) and scaling the project for the given circumstances. The project workplan should focus on the tasks included in the OUM Implement Core Workflow, which identifies the core tasks within the Implement Focus Area.

Understanding and incorporating Scrum principles and tools in an OUM project is possible and may improve delivery. Both approaches are compatible and can complement each other when applied correctly.

References: Agile Manifesto - http://www.agilemanifesto.org, *Agile & Iterative Development: A Manager's Guide* by Craig Larman, and *Balancing Agility and Discipline: A Guide for the Perplexed* by Barry Boehm and Richard Turner

References and Works Cited

- Takeuchi and Nonaka. a. (n.d.). http://harvardbusinessonline.hbsp.harvard.edu/b01/en/common/item_detail.jhtml?id=86116.
- Schwaber, Ken. 2004. *Agile Project Management with Scrum*. Redmond, WA.: Microsoft Press
- Larmen, Craig. 2004. *Agile & Iterative Development; A Managers Guide*. Boston, MA.: Pearson Education, Inc.
- Bittner, Kurt. 2007. *Managing Iterative Software Development Projects*. Boston, MA: Pearson Education, Inc.
- Boehm, B. and R. Turner. 2004. *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston, MA: Addison-Wesley
- Cockburn, A. 2002. *Agile Software Development*. Boston, MA: Addison-Wesley.
- Cohn, Mike, 1962 -. *Agile estimating and Planning*, Upper Saddle River, NJ: Pearson Education, Inc.
- A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fourth Edition, Newton Square PA: Project Management Institute, Inc.
- http://www.agilemanifesto.org.
- http://en.wikipedia.org/wiki/Scrum_%28development%29
- http://alistair.cockburn.us
- http://www.controlchaos.com/
- http://martinfowler.com/articles/itsNotJustStandingUp.html
- http://Agilesoftwaredevelopment.com/scrum
- http://www.mountaingoatsoftware.com
- http://scrum.alliance.org

Managing an OUM Project Using Scrum
February 2011

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com